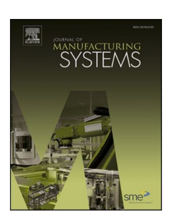
Journal of Manufacturing Systems 78 (2025) 178–186

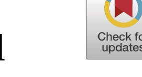
Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys



Technical paper



Material removal rate optimization with bayesian optimized differential evolution based on deep learning in robotic polishing

Ruoxin Wang [a], Chi Fai Cheung [a,*], Yikai Zang [b], Chunjin Wang [a], Changlin Liu [a,*]

[a] State Key Laboratory of Ultra-precision Machining Technology, Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

[b] State Key Laboratory of Intelligent Manufacturing Equipment and Technology, School of Mechanical Science & Engineering, Huazhong University of Science and Technology, Wuhan, China

ARTICLE INFO

Keywords:
Robotic polishing
Ultra-precision Machining
Differential evolution
Bayesian Optimization
Material removal rate

ABSTRACT

Large aperture aspheric optical surfaces (LAAOS) have been applied in many industries, but their high requirements for precision and efficiency make manufacturing more challenging. Robotic polishing is a representative computer-controlled optical surfacing technique to manufacture LAAOS with low-cost and high-efficiency. However, how to achieve the highest material removal rate (MRR) involves many process parameters. It is difficult to determine the optimal parameter settings since the complex relationships among them. In this paper, a novel Bayesian optimized differential evolution based on deep learning method is proposed to optimize the MRR, in which the designed deep neural network is responsible for MRR modeling and Bayesian optimized differential evolution is used for MRR optimization. Bayesian optimization is used to find the best hyperparameter of differential evolution method so as to improve optimization performance. To evaluate the proposed method, a series of robotic polishing experiments are conducted to build the MRR model. The optimization performance comparison experiments show the superiority of our proposed method, which increases MRR by an average of 0.16.

1. Introduction

The large aperture aspheric optical surfaces have been applied in various scenarios, such as large-aperture astronomical telescopes [1], X-ray mirrors for synchrotron radiation [2] and high-NA optics in EUV lithography [3] since the applications of aspheric surfaces [4] can achieve better imaging performance with fewer optical components and lower weight. However, processing becomes more challenging as aspheric optical component size and accuracy requirements increase. To improve the precision and efficiency of the large aperture aspheric optical surfaces manufacturing, a series of polishing techniques have been developed, such as magnetorheological finishing [5], ion beam figuring [6], bonnet polishing [7], fluid jet polishing [8], stress lap polishing [9] and computer-controlled optical surfacing (CCOS) [10]. Since CCOS can achieve low-cost mass production of large-aperture optical surfaces with high polishing efficiency and shape accuracy, it has attracted much research attention. As a kind of computer-controlled optical surfacing (CCOS) and ultra-precision machining technology, robotic polishing (RP) is a deterministic, sub-aperture polishing technology applicable to efficient and high-quality processing of large aperture aspherical optical components. During the processing of RP, a dual-rotor planetary polishing tool driven by an industrial robot to dwell on the surface of workpiece to achieve quantitative material removal rate (MRR). Specifically, the polishing pad works in a double planetary motion driven by the ball-head shaft, the spinning and orbital motion of the polishing pad are driven by two independent motors. The contact mode between polishing pad and workpiece is flexible, the polishing pressure is regulated by two micro cylinders mounted on the upper part of the spindle. The radius of orbital motion is adjusted by an eccentric device which installed between the orbital motor and spinning motor. The advantage of dual-rotor planetary polishing is that it can generate Gaussian-shaped polishing footprint, which could contribute to the effective convergence of form error. Since the MRR value is influenced by various parameters, such as the ratio of spinning and orbiting, radius of orbiting, pressure between polishing pad and workpiece, dwell time, property of slurries etc., the relationships between MRR and process parameters is so complex to model. Therefore, it is difficult to determine the suitable parameters to achieve the required MRR. In addition, since ratio of

* Corresponding authors.
E-mail addresses: benny.cheung@polyu.edu.hk (C.F. Cheung), changlin.liu@polyu.edu.hk (C. Liu).

https://doi.org/10.1016/j.jmsy.2024.11.014
Received 22 September 2024; Received in revised form 7 November 2024; Accepted 18 November 2024
Available online 29 November 2024
0278-6125/

spinning and orbiting, radius of orbiting and pressure determine the removal function profile and removal rate characteristics, and these three parameters have an important effect on a stable and ideal Gaussian removal function, which is a key step in deterministic material removal. However, traditionally, a series of experiments need to be conducted to find the best parameter settings so as to achieve desirable performance, which is time-consuming and energy wasting.

In this paper, a novel Bayesian optimized differential evolution based on deep learning method (BodeDL) is proposed to obtain the optimal parameters of RP, including pressure (P), ratio of spinning and orbiting (RSO) and radius of orbiting (R) with maximizing MRR. In this method, a deep neural network is designed to build the model between MRR, and process parameters of RP based on a series of experimental data. The mean absolute prediction error and mean standard deviation of MRR can reach 0.666 and 0.496, respectively. A Bayesian optimized differential evolution (BoDE) algorithm is developed to solve MRR optimization problem, in which the hyperparameters of the differential evolution algorithm, including scale factor (F) and crossover operator (CR), are determined by Bayesian optimization (BO) so as to avoid repetitive hyperparameter selection process. Compared with other methods, BoDE increases MRR by an average of 0.12, and the average ΔMRR of BoDE compared with DE reaches 0.16.

2. Related work

2.1. Optimization in robotic polishing

Recently, many robotic polishing process optimization methods have been proposed to achieve efficient and green manufacturing. Cao et al. [11] used polynomial function to model the task energy consumption characteristic of RP and achieved prediction error of less than 7 %. And then an adaptive genetic algorithm with elite retention strategy was utilized to optimize the RP process to minimize the energy consumption. Xiao et al. [12] first developed a mathematic model to predict pressure distribution and the material removal depths and then proposed a novel high-priority subregion searching algorithm to optimize path direction, tool dwell times, and the path spacings so as to achieve desired material removal depths. Pan et al. [13] proposed an optimization method to improve robot static stiffness performance based on the analysis of influence of robot system stiffness on polishing area size and removal. The optimization performances, including work stiffness and polished surface roughness, were evaluated through simulation and experiments. Ding et al. [14] aimed to stabilize actuator contact force through applying reinforcement learning algorithm. They took three influence parameters into consideration to analyze the impedance performance through numerical simulation method and obtained the best parameters. The polishing experiments of turbine blades were conducted to validate the superiority of the proposed method. Huang et al. [15] illustrated a dwell time optimization method to improve the polishing quality and efficiency. A monotone projection gradient method and cubic B-spline interpolation method were adopted, and simulation and experiments were conducted to show that methods improve the solution accuracy of dwell time and convergence rate of form error. Therefore, RP process optimization is a very worthy and necessary research topic.

2.2. Intelligent optimization of material removal rate

Since material removal rate (MRR) is an important characteristic in various processes, there are many studies focused on building MRR model to conduct MRR optimization. Since the rapid development of intelligent optimization algorithms, many studies applied them to MRR optimization. Alam et al. [16] used response surface methodology to build the model of MRR and cutting parameters of a high-speed machining process, including cutting speed, feed rate and depth of cut. Genetic algorithm is adopted to obtain the optimal cutting parameters with the objective of maximizing the MRR. Sharma et al. [17] investigated the effect of parameters of wire-cut electrical discharge machining, including pulse on-time, pulse off-time and servo voltage on MRR. A hybrid algorithm is proposed to optimize MRR through combining grey approach and harmony search. Li et al. [18] proposed an improved teaching–learning-based optimization method to achieve multi-objective optimization in milling process with three objective functions, including MRR, surface roughness and machining energy consumption. Rouniyar et al. [19] aimed to optimize magnetic field assisted powder mixed EDM process through combining artificial neural network (ANN) and genetic algorithm (GA). A feed-forward back-propagation ANN was utilized to build model between MRR and machining parameters and the optimal parameters, including peak current, spark on duration, spark off duration, magnetic field and powder concentration were obtained through GA. Osorio-Pinzon et al. [20] determined the optimal cutting parameters with a multi-objective particle swarm optimization (MOPSO) method. The ANN model was built through the data collected from numerical simulations of Aluminum 6063 machining. The pareto optimal solutions were obtained through maximizing MRR, maximizing microstructure refinement, and minimizing cutting force using MOPSO. Therefore, optimizing MRR is very important to improve the performance and efficiency of RP.

2.3. Differential evolution algorithms

Optimization aims to determine the best elements to minimize or maximize the objective functions under some constraints. Meta heuristic algorithms have been applied in many real-world problems, since they can obtain good solutions through less computational effort than other algorithms, such as optimization algorithms and iterative algorithms. In meta metaheuristic algorithms, evolutionary algorithms (EAs) mimic the behaviors of evolution and living organisms and are capable of handling high dimensional problems with good robustness to noise.

As a representative EA, differential evolution (DE) was first proposed by Storn in 1996 [21] to solve global optimization in a continuous domain. Then its variants sprung up in recent years since they have better performance in terms of convergence speed and solution stability [22]. They aim to improve DE performance through developing parameter selection methods and mutation strategies. jDE was proposed by Brest et al. [23] to implement self-adaptive DE through introducing four parameters to calculate scalar factor (F) and crossover operator (CR) adaptively. They achieved 1st rank of IEEE Congress on Evolutionary Computation (CEC competition) in 2009. Tanabe et al. [24] illustrated an improved JADE [25] algorithm named SHADE to eliminate its greediness. They adopted a historical memory technique to store information of F and CR in each generation to guide the parameter setting. To alleviate computational costs of SHADE, Tanabe and Fukunaga [26] developed a new DE algorithm named LSHADE by introducing a linear population strategy. It achieved the best performance in 2014 CEC competition with low computational burden. Due to its competitiveness, many related algorithms were proposed, such as LSHADE-Epsin [27], LSHADE-cnEpsin [28] and LSHADE-RSP [29]. Apart from the good performance of DE, DE has many advantages in its industry application. Firstly, the DE mutation is the first functioning operator in the evolutionary process. The mutation affects all individuals in the present population, allowing for more variability in the search process. Secondly, compared with other EA algorithms, DE converges faster, which can make optimization process faster. Therefore, it is more suitable for online optimization during manufacturing process. Thirdly, the parameter setting is easier, which makes the results more stable. Therefore, DE is a very potential solution to solve MRR optimization problems.

Although many studies focused on RP process optimization, little attention is paid to MRR optimization in the RP process since MRR is difficult to model due to complex parameter relationships. Therefore, a MRR model building is necessary for further MRR optimization. In addition, the hyperparameter settings of DE are difficult to determine

and important to final optimization performance. In this paper, we proposed a novel DE method to optimize a designed deep learning-based MRR model, in which Bayesian optimization determines the hyperparameters of DE to obtain the better MRR in RP process. The proposed method can support parameter settings in RP process and facilitate the development of the digital twin RP process [30].

3. MRR optimization based on BodeDL

The structure of the proposed method, BodeDL, is shown in Fig. 1. The method is composed of two main modules, including MRR modeling and MRR optimization which are responsible for building MRR model of RP based on deep neural network learning and optimizing MRR based on Bayesian optimized differential evolution algorithm. In MRR modeling module, a model of MRR in RP process is established considering three process parameters, including pressure (P), ratio of spinning and orbiting (RSO) and radius of orbiting (R). Firstly, the dataset is collected through conducting a series of RP process experiments with different process parameter settings. Then data preprocessing techniques, such as max-min normalization and B-spline transformation, are adopted to reduce the value range differences among different parameters and avoid missing interaction features. The transformed features are passed into a designed deep neural network (DNN) for learning the relationships between features of process parameters and MRR. The trained model serves as the objective function of optimization problem, ready for MRR optimization module to solve. In the MRR optimization module, a Bayesian optimized differential evolution method is illustrated to optimize MRR.

The Bayesian optimization aims to obtain the best hyperparameters of DE, including scale factor (F) and crossover rate (CR) of DE. DE is responsible to maximize the MRR model so as to obtain the best process parameters. In the Bayesian optimization algorithm, initial points are randomly generated and passed into DE algorithm to set hyperparameters (F and CR). Then the population is initialized, and fitness function is calculated. More specifically, the population is regarded as test samples of trained DNN model so that fitness function calculation means DNN prediction. After that, three evolutional operators, including mutation, crossover and selection are performed if the stopping criterion is not met. When the optimization process stops, the best individual and best objective value can be obtained. Then the best objective value is passed back to Bayesian optimization algorithm for evaluation. Then the Gaussian process regression is performed to create surrogate model. If the stopping criterion is not met, acquisition function is optimized. A query point will be selected to pass to DE and a new iteration begins until they reach the stopping criterion. The best hyperparameters are obtained, and optimal MRR and process parameters are further obtained.

3.1. Data preprocessing

Data normalization is a common technique for converting data from inconsistent scales to a single scale, allowing the model to converge faster and produce more accurate results. It decreases data variance to some extent. Normalizing techniques include zero-mean normalizing, log function conversion normalization, and quantile normalization. Based on the effectiveness of numerous normalizing approaches on our data, Max-Min normalization is chosen for data normalization in this module to limit the effects of distinct data characteristics with varying units or magnitudes. Min-max normalization (i.e., rescaling) can be used to rescale the range of data features to scale the range in [0,1] depending on the nature of the data, which can be defined by

$$p_{normalization} = \frac{p - p_{\min}}{p_{\max} - p_{\min}} \tag{1}$$

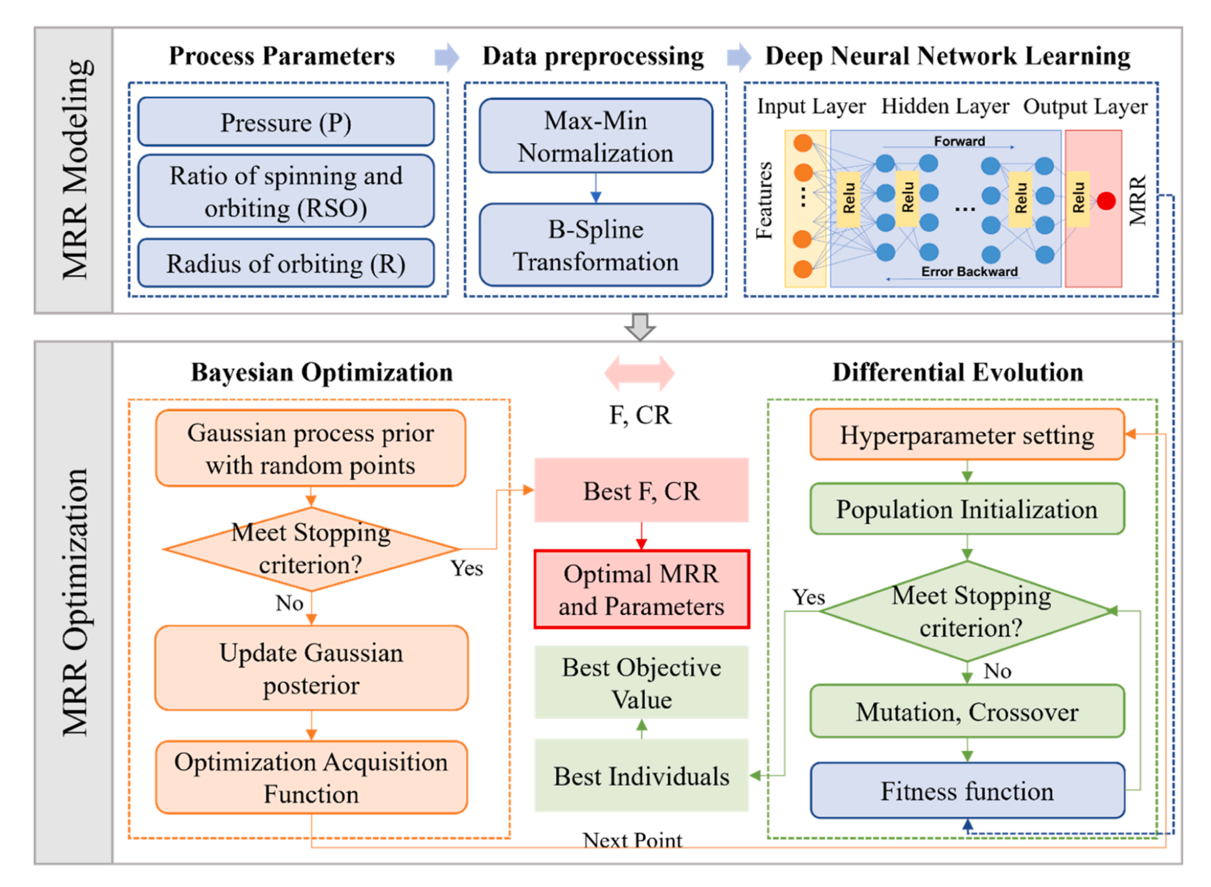


Fig. 1. The structure of Bayesian optimized differential evolution based on deep learning method.

Table 1
The structure of deep neural network.

Layer	Structure	Layer	Structure
Input Layer	Input - $(k + d - 1)$	Hidden Layer−6,7	FC−16-Relu
Hidden Layer−1	FC−128-Relu	Hidden Layer−8,9	FC−8-Relu
Hidden Layer−2, 3	FC−64-Relu	Hidden Layer−10,11	FC−4-Relu
Hidden Layer−4, 5	FC−32-Relu	Output Layer	FC−1

where p denotes the parameters of RP, $p_{normalization}$ denotes normalized input data, p_{min} denotes the minimal value of parameters and $p_{\max}$ is the maximal value of parameters.

One of the well-known methods for data mining is feature transformation, which is a crucial stage in feature engineering [31]. To make the representation of the data more suited to process for the downstream analysis, it translates a collection of values for a feature to a new set of values. There are many transformation techniques, such as polynomial transformation, spline transformation and function transformation, such as power transformation and quantile transformation. Spline transformation can generate univariate B-spline bases for features. It has many advantages compared with polynomial transformation and function transformation. Firstly, B-splines are very flexible and robust. Secondly, B-splines, unlike polynomials, do not exhibit oscillatory behavior at their borders. This is known as Runge's phenomenon. Thirdly, B-splines are useful for extrapolation beyond the limits, that is, outside the range of fitted values. Fourthly, B-splines generate a feature matrix with a banded structure, resulting in a matrix with good numerical properties, e.g., a low condition number, in sharp contrast to a matrix of polynomials, which goes under the name Vandermonde matrix. A B-spline with degree $d+1$ is a set of piecewise polynomial functions $B_{i,d}(k)$. Knot vector, K= $\{k_0, k_1, \ldots, k_m | k_0 \le k_1 \le \cdots \le k_m\}$, is a set of polynomial pieces intersect points. The B-spline of degree $d = 0$ and $d > 0$ are defined as Eq. (2.1) and Eq. (2.2), respectively.

$$B_{i,0}(x) := \begin{cases} 1 & \text{if } k_i \le k \le k_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

$$B_{i,d}(k) := \frac{k - k_i}{k_{i+d} - k_i} B_{i,d-1}(k) + \frac{k_{i+d+1} - k}{k_{i+d+1} - k_{i+1}} B_{i+1,d-1}(k) \tag{2.2}$$

where $B_{i,d}(k) = \begin{cases} \text{non} - \text{zero} & \text{if } k_i \le k \le k_{i+d+1}, \\ 0 & \text{otherwise.} \end{cases}$ and $\sum_{i=0}^{m-d-1} B_{i,d}(k) = 1$.After spline transformation, *BSplineTran*, we can get a new features with $k+d-1$ dimensions when setting knot to k and degree to d.

3.2. BodeDL method for MRR optimization in robotic polishing

3.2.1. Deep learning based MRR modeling

To achieve the MRR optimization, the relationship model between RP process parameters and MRR has to be established first. Since the neural network has powerful ability to model complex relationships, in this paper, a deep neural network, *DNN*, is designed to build a MRR predictive model with R experiments data, $D = \{(p_i, y_i), \ldots, (p_r, y_r)\}$. It has 11 hidden layers with RELU activation function, the detailed layer information is listed in Table 1, where the structure of each layer is represented as "(layer type) - (number of neurons) - (activate function)". Nesterov-accelerated Adaptive Moment Estimation (Nadam), whose updated rule is defined in Eq. (3), is used to be an optimizer, which is an enhanced version of Adam through adding Nesterov momentum and the loss function is the mean absolute error of predicted value, $\hat{y}$, and real value, y,which is defined in Eq. (6), where DNN_D denotes the trained *DNN* model on data D.

Table 2
Experimental parameter setting and MRR.

#Experiments	RSO (direction)	RSO (value)	P (Mpa)	R (mm)	MRR (0.001 *mm^3/ 3 min)
1	1	0	0.1	2	7.39
2	0	0.05	0.1	2	10.11
3	0	0.1	0.1	2	9.32
4	0	0.25	0.1	2	11.55
5	0	0.5	0.1	2	9.95
6	0	0.75	0.1	2	12.2
7	0	1	0.1	2	11.12
8	1	0.1	0.1	2	11.66
9	1	0.25	0.1	2	11.77
10	1	0.5	0.1	2	12.37
11	1	0.75	0.1	2	11.02
12	1	1	0.1	2	11.34
13	0	0.05	0.06	2	6.27
14	0	0.05	0.08	2	8.53
15	0	0.05	0.12	2	13.52
16	0	0.05	0.15	2	15.67
17	0	0.05	0.16	2	16.44
18	0	0.05	0.18	2	17.53
19	0	0.05	0.19	2	16.53
20	0	0.05	0.2	2	17.98
21	0	0.05	0.21	2	21.46
22	0	0.05	0.22	2	20.87
23	0	0.05	0.16	0.5	4.55
24	0	0.05	0.16	1	4.55
25	0	0.05	0.16	1.3	11.49
26	0	0.05	0.16	1.6	12.96
27	0	0.05	0.16	2.3	18.77
28	0	0.05	0.16	2.5	18.09
29	0	0.05	0.16	3	14.14
30	0	0.05	0.16	4	17.58
31	0	0.05	0.16	5	19.27

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \left(\beta_1 \hat{m}_{t-1} + \frac{(1 - \beta_1) g_t}{1 - \beta_1^t} \right) \tag{3}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \qquad v_t = \beta_2 v_{t-1} + (1 - \beta_2) {g_t}^2 \tag{4}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \qquad m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{5}$$

where θ_t is parameters at time step t; η is learning rate; g_t is gradient at time step t; β_1, β_2 are decay rates; v_t is the exponentially decaying average of past squared gradients; m_t is the exponentially decaying average of past gradients; and $\hat{v}_t$ and $\hat{m}_t$ are the bias-corrected estimate of v_t and m_t, respectively, which is defined in Eq. (4) and Eq. (5). ϵ is a smoothing term that avoids division by zero.

$$loss = \frac{1}{K} \sum_{i=0}^{K} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad s.t. \tag{6}$$

$$\hat{y}_i = DNN_D(p_i)$$

3.2.2. Bayesian optimized differential evolution based MRR optimization

In practice, how to set the optimal RP parameters to obtain the best MRR is a challenging issue. Therefore, in this paper, a Bayesian optimized differential evolution method is proposed to optimize the MRR model with the objective of maximizing MRR. In the Bayesian Optimization method, the minimum or maximum of an objective function is sought by using the Bayes Theorem to guide the search.

The Bayesian optimization aims to solve the problem shown in Eq. (7). Here $f(x)$ is (unknown) response value of objective function at location x, we use f for simplicity. And y denotes the actual value at x, which can be represented as Eq. (8), where ε is a noise term, and σ_n is the variance which can be estimated from the data. In the procedure of

Bayesian optimization, there are two main parts, objective function modeling using a surrogate function and sampling by optimizing an acquisition function.

$$x^* = \underset{x \in A}{\operatorname{argmax}} f(x) \tag{7}$$

$$y = f(x) + \varepsilon \quad s.t. \quad \varepsilon \sim \mathcal{N}(0, \sigma_n{}^2) \tag{8}$$

3.2.2.1. Objective function. The objective function is the black box model, DE, which can output the optimal MRR with regard to initial population, P. The objective function can be represented as Eq. (9) and the *De* method is shown in Algorithm I.

$$f = De(x|DNN_D(P)) \quad s.t. \quad x = [CR, F] \text{ and } \mathrm{CR} \in (0,1), F \in (0,1) \tag{9}$$

Algorithm I. DE based on deep learning.

Input: Population size |P|, F, CR, Maximum generations $MaxGen$, Dimension Dim, Upper boundaries U, Low boundaries L.

Output: Optimal objective MRR.

Begin

Randomly initialize $|P|$ individuals, $L_{i,j} \le P_{i,j}^0 \le U_{i,j}, i \in (1,|P|), j \in (1, Dim)$, g =0

while $g < MaxGen$ do

for $i \leftarrow 1$ to |P| do

randomly chose three individuals P_{r1}^g, P_{r2}^g, and P_{r3}^g

$V_i^{g+1} = P_{r1}^g + F(P_{r2}^g - P_{r3}^g)$

random chose a value $r_{i,j} \in (0,1)$

if $r_{i,j} \le CR$ then

$U_{i,j}{}^{g+1} = V_{i,j}^{g+1}$

else

$U_{i,j}{}^{g+1} = P_{i,j}^g$

$B_U_i^{g+1} = BSplineTran(U_i^{g+1})$

$B_P_i^g = BSplineTran(P_i^g)$,

if $DNN_D(B_U_i^{g+1}) \ge DNN_D(B_P_i^g)$ then

$P_i^{g+1} = U_i^{g+1}$

else

$P_i^{g+1} = P_i^g$

g=g+1

return $DNN_D(B_P_i^{g+1})$

End

3.2.2.2. Surrogate function. The surrogate function is a technique to approximate the mapping of input data to an output score and gaussian process regression is a perfect model to approximate the distribution of many variables with gaussian distributions. Assuming the observed data $X = [x_1, \ldots, x_n]$, the probability distribution of f follows a Gaussian distribution with a zero mean can be represented as Eq. (10). In this paper, a gaussian process (GP) model with Matern kernel defined in Eq. (11), is adopted. Matern kernel is an extension of radial basis function (RBF) kernel through adding a parameter ν to controls the smoothness of final function. Here we set ν to 2.5 to make function twice differentiable.

$$f \sim \mathcal{N}(0, K(X,X)) \tag{10}$$

where $\mu(X) = [\mu(x_1), \ldots, \mu(x_n)]$ are the mean vector, $f = [f(x_1), \ldots, f(x_n)]$ and $K(X,X) = \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_n) \\ \cdots & \cdots & \cdots \\ k(x_1,x_1) & \cdots & k(x_n,x_n) \end{bmatrix}$.

$$k(x_i,x_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{l} d(x_i,x_j) \right)^{\nu} \mathrm{K}_\nu \left(\frac{\sqrt{2\nu}}{l} d(x_i,x_j) \right) \tag{11}$$

where K_ν is modified Bessel function; Γ is gamma function; $d(x_i,x_j)$ is the Euclidean distance, l is a length-scale parameter. Then the joint distribution of the observed value y and response value f_* test points X_* is shown Eq. (12).

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n{}^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{12}$$

Then using the Bayes theorem, the posterior of f_* can be obtained represented in Eq. (13).

$$f_*|X_*, X, y \sim \mathcal{N}(\overline{f_*}, \mathrm{cov}(f_*)), \quad s.t.$$

$$\overline{f_*} := K(K_*, X)\left[K(X,X) + \sigma_n{}^2 I\right]^{-1} y$$

$$\mathrm{cov}(f_*) = K(X_*, X_*) - K(X_*, X)\left[K(X,X) + \sigma_n{}^2 I\right]^{-1} K(X, X_*) \quad (13)$$

In this paper, the upper confidence bounds (UCB) method serves as acquisition function, which can be represented as Eq. (14).

$$a(x;\lambda) = \mu(x) + \lambda\sigma(x) \quad (14)$$

where $\mu(x)$ and $\sigma(x)$ are the mean value and standard deviation of GP.

4. Experimental settings and results

4.1. Dataset

To build and validate our proposed model, 31 RP experiments were conducted with different parameter settings, including direction of RSO (1 is same direction and 0 is opposite direction), value of RSO, P, and R. The value of parameters and corresponding MRR are listed in Table 2. Here, MRR represents the material removal rate per unit time, and its unit in this paper is 0.001 *mm^3/3 min, that is, the material removal volume under a certain set of process parameters within 3 min. In all experiments, 10 mm diameter asphalt polishing pads were used. Since asphalt as a flexible pad can adapt to the changes in the polishing surface shape within a certain pressure range, it will cause irreversible deformation if the pressure exceeds some range, resulting in distortion of the removal function, which is not conducive to error convergence. Therefore, the range of P was set from 0.06Mpa to 0.22Mpa. The range of R was set from 0.5 mm to 5 mm since if the R is greater than 5, the central area of the removal function does not contact the polishing pad, resulting in a removal amount of 0 in the central area and a V-shaped removal function, which is not ideal removal function for polishing. For the parameter RSO, two working conditions, including the revolution and rotation speeds were opposite and the same, were considered. Since the maximum speed of the revolution and rotation motor was designed to be 200 r/min. Therefore, the maximum rotation speed is 200 r/min and the corresponding RSO is calculated to analyze the influence of RSO on the MRR.

The detailed robotic polishing experimental setting is shown in Fig. 2. Fig. 2(a) shows the RP equipment which includes a self-developed dual-rotor planetary polishing tool, a KUKA KR70 industrial robotic arm, and a rotary table. Fig. 2(b) is the workpiece with a diameter of 180 mm, the detailed dwell spot setting is shown in Fig. 2(e). The polishing pad polish the workpiece with the tool path shown in Fig. 2(d) under the specific polishing pressure. During the polishing process, the polishing pad rotates as Fig. 2(c), where ω_q is speed of orbiting, ω_p is speed of speed of spinning, oo' is radius of orbiting and r is diameter of polishing pad. The dwelling time of all experiments is 180 s. The polishing slurry is a Al2O3 suspension (AlumTX 1000 nm, Universal Photonics, United States), the concentration is 20–30 wt%, the average size of the abrasive grains is 1μm and the PH is 4.0–7.0. The surface profile was measured by LUPHOScan 420HD (as shown in Fig. 3(a)) to calculate the volume of material removal. All experiments were repeated three times, and MRR is the average of these three times. Fig. 3(b) shows the material removal footprints of some experiments from top to bottom, including Experiment No. 1, Experiment No. 2, and Experiment No. 3. The footprints from left to right in Fig. 3(b) are the results of three repeated experiments.

4.2. Implementation details

To establish MRR prediction model, 90 % and 10 % of data serve as

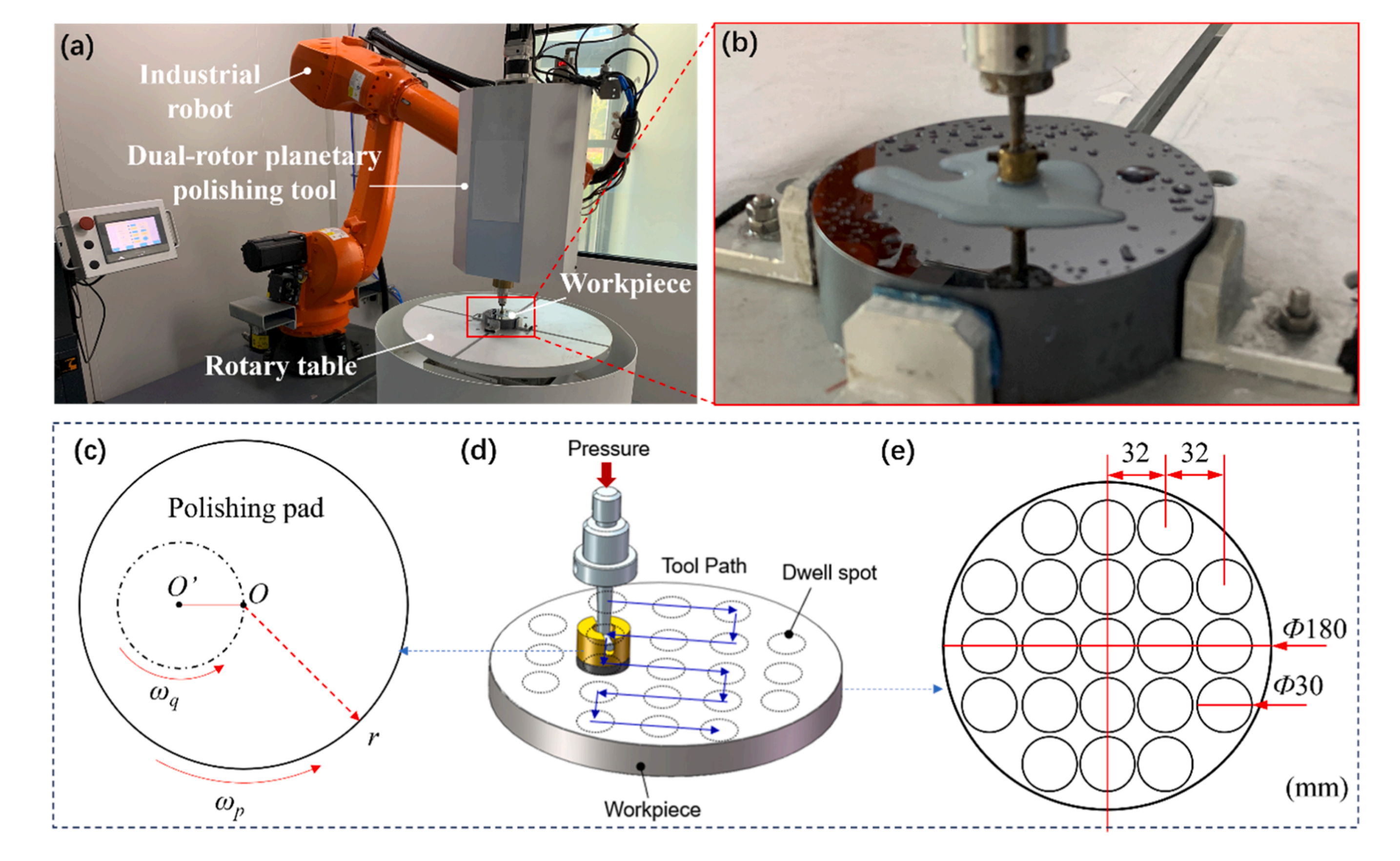


Fig. 2. The robotic polishing experimental settings. (a) RP equipment; (b) Workpiece with a diameter of 180 mm; (c) Mechanism of polishing pad rotation. (d) Polishing pad tool path during RP process; (e) Dwell spot setting.

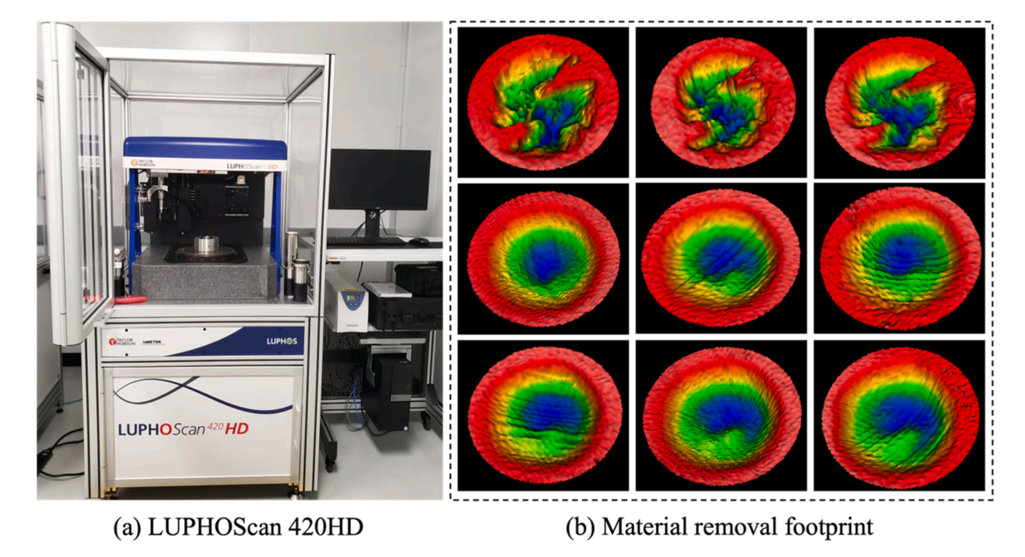


Fig. 3. The material removal measurement results.

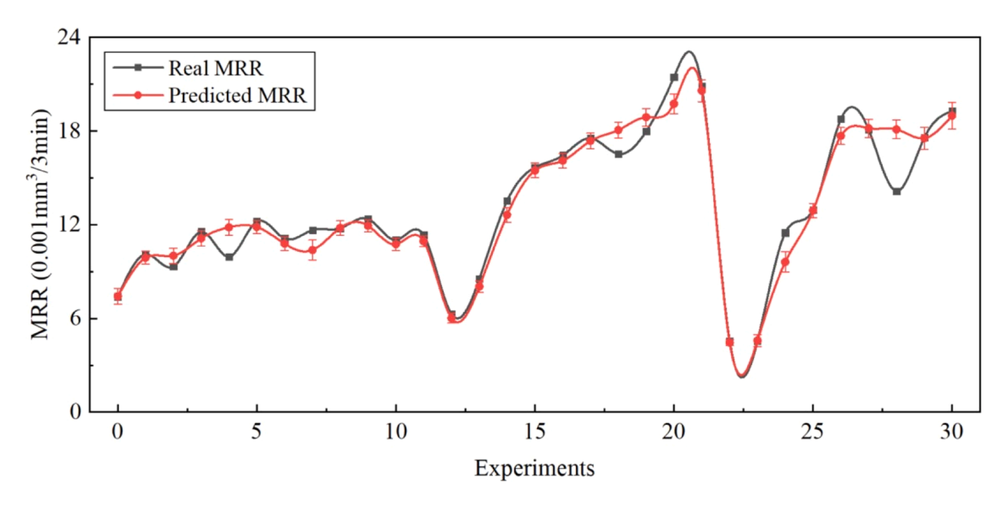


Fig. 4. The comparison between real MRR and predicted MRR.

Table 3
The MRR modeling performance with different B-spline Transformation.

Degree	Number of knots	Feature dimension	MAE
0	0	4	1.889
1	2	8	2.228
1	3	12	2.627
2	3	16	**1.780**
2	4	20	2.432
3	4	24	1.844
3	5	28	2.495
4	5	32	1.963

training and validation dataset to train the designed DNN model. Training epoch is set to 200, and batch size is set to 1. The training experiments were conducted ten times. The mean average error (MAE) is used as evaluation metric. For the MRR optimization model, the initial population size of DE is 20, and the maximum evolution generation of DE is set to 30. The number of iterations of BO is set to 25. All experiments run on a computer with an i7–9750H 2.60 GHz CPU and NVIDIA GEFORCE RTX 2060 GPU. The code was implemented in python.

4.3. Results of MRR modeling based on deep learning

To show the performance of deep learning model, we compare the real MRR and predicted MRR of each RP experiment. The predicted MRR is an average of ten times results with standard error. The comparison results in Fig. 4 show that our method can model MRR perfectly in most of the experiments [32,33]. For example, the differences between predicted MRR and measured MRR in experiment No. 1, 9 and 24 are less than 0.03. In addition, the differences between predicted MRR and measured MRR in 65 % experiments are smaller than 0.5. The mean absolute error and deviation of 31 experiments remains small with an average of 0.666 and 0.496, respectively. To explore the performance of deep learning model, we further compared the prediction performance on different feature dimensions. The results listed in Table 3, where different feature dimensions are obtained with different B-spline Transformation settings. The results show that the best feature dimension is 16 when degree is set to 2 and number of knots is set to 3. These results mean that the feature dimension plays an important role in prediction performance. The main reason is that the feature dimension represents the amount of parameter information for final MRR prediction. In addition, larger feature dimensions do not guarantee better performance. The reason is that if too much information is provided, there is some redundant information that cannot help model accuracy but even hurt the performance like noisy data. Furthermore, the larger

Table 4
The performance comparison of DE and BoDE.

Methods	F	CR	MRR (DE)	MRR (BoDE)	ΔMRR
DE/best/1/bin	0.3392	0.2749	22.56	22.59	0.03
DE/best/1/L	0.3968	0.5388	22.54	22.59	0.05
DE/rand/1/bin	0.8441	0.6097	22.51	22.59	0.08
DE/rand/1/L	0.88441	0.6097	22.55	22.59	0.04
DE/targetToBest/1/bin	0.4152	0.6845	22.19	22.59	0.40
DE/targetToBest/1/L	0.9214	0.8171	22.56	22.59	0.03
DE/current-to-best/1/bin	0.3941	0.5407	22.25	22.59	0.34
DE/current-to-best/1/L	0.997	0.2134	22.27	22.59	0.32
Average					0.16

degree achieves better performance with the same number of knots, and the smaller number of knots achieves better performance with the same degree. The reason is that the larger the degree, the more process parameter information is to be introduced, and the smaller the number of knots, the less unknown function is to be determined.

4.4. Results of MRR optimization based on BoDE

Based on the MRR prediction model, the proposed optimization method, BoDE, can be used to obtain the best RP parameters to yield the largest MRR. To explore the performance of Bayesian optimization (BO) in the DE, we conducted experiments on the most popular DE variants, such as DE/best/1/bin, DE/best/1/L, DE/rand/1/bin, DE/rand/1/L, DE/targetToBest/1/bin, DE/targetToBest/1/L, DE/current-to-best/1/bin, DE/current-to-best/1/L. Here, best/1, rand/1, targetToBest/1 and current-to-best/1 are different mutation strategies. L and bin are different crossover strategies, i.e., exponential crossover operator and Binomial distribution crossover operator. The best MRR obtained by DE and BoDE are listed in Table 4. Compared with DE and BoDE, the results show that BO can improve the MRR for all DE variants with the average MRR increase of 0.16. In addition, after optimized by BO, MRR of all DE variants all can yield best MRR with different F and CR. The reason is that all these DE variants' performance is dependent on the F and CR settings. In addition, since they use different crossover and mutation strategies, the F and CR settings are also different.

To further explore the BoDE performance, we compared the optimization performance with state-of-the-art methods, such as genetic algorithm (GA) [34] and GA variants, including elite reservation GA (EGA) [35], strengthen elitist GA(SEGA) [36], generational gap GA (GGAP_GA) [37], stud GA (StudGA) [38] and steady state GA (steadyGA) [39], evolutionary strategy (ES) [40] and DE. The best MRR achieved by these methods are listed in Table 5. The results show that BoDE obtained the best MRR than other methods, with an average ΔMRR of 0.12, which means that BO plays a more important role in DE algorithms and the performance of DE is more sensitive to hyperparameter setting. And compared with GA and ES methods, DE shows its superiority. The main reason is that the DE mutation is the first functioning operator in the evolutionary process. The DE mutation affects all individuals in the present population, allowing for more variability in the search process.

Table 5
The performance comparison of BoDE with other state-of-the-art methods.

Methods	MRR	ΔMRR
GA	22.57	0.02
SEGA	22.18	0.41
EGA	22.50	0.09
GGAP_GA	22.47	0.12
studGA	22.53	0.06
steadyGA	21.52	0.07
ES_1_plus_1	22.55	0.04
ES_miu_plus_lambda	22.31	0.28
DE (best/1/bin)	22.56	0.03
BoDE	**22.59**	(Average)0.12

5. Conclusion

In this paper, a novel Bayesian optimized differential evolution based on deep learning method is proposed to maximize material removal rate (MRR) of robotic polishing so as to improve the polishing efficiency which includes two main modules, namely MRR modeling and MRR optimization. MRR modeling aims to predict the MRR according to RP process parameters based on deep neural network. Then the Bayesian optimized differential evolution (BoDE) method is used to optimize MRR, where Bayesian optimization is adopted to obtain the optimal hyperparameters of differential evolution, including mutation scale factor and crossover rate. Then the best parameters of RP and MRR can be obtained. To build the MRR model, a serial of RP experiments was performed to train deep learning model. Based on MRR model, optimization comparison experiments were conducted to validate the proposed model performance. The results show that BoDE can increase MRR by an average of 0.16 compared with other DE variants. In addition, BoDE can achieve better performance than GA variants with an average improvement of 0.12.

Although the proposed method can achieve good prediction performance, many factors in the RP process can be explored in future work to improve the performance of the proposed model, such as more polishing parameters, in-process signals, a larger dataset, and other response variables. Then, the model can be applied to an online monitoring system to provide real-time RP performance prediction. In addition, by adding a signal transmission module and other control modules, the robot can automatically adjust the process parameters according to the optimization results to achieve closed-loop control. This will help to accelerate the development of digital twin robotic polishing.

CRediT authorship contribution statement

Changlin Liu: Writing – review & editing, Resources, Data curation. **Chunjin Wang:** Writing – review & editing, Resources. **Yikai Zang:** Visualization, Data curation, Conceptualization. **Chi Fai Cheung:** Writing – review & editing, Supervision, Resources, Funding acquisition. **Ruoxin Wang:** Writing – original draft, Validation, Methodology, Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work described in this paper was mainly supported by a grant from the Research Grants Council (Project No. R5047–22) and Innovation and Technology Commission (ITC) (Project No.: MHP/151/22) of the Government of the Hong Kong Special Administrative Region, China.

References

[1] Graves LR, Smith GA, Apai D, Kim DW. Precision optics manufacturing and control for next-generation large telescopes. Nanomanuf Metrol 2019;2:65–90.
[2] Thiess H, Lasser H, Siewert F. Fabrication of X-ray mirrors for synchrotron applications. Nucl Instrum Meth A 2010;616(2-3):157–61.
[3] Wischmeier L, Graeupner P, Kuerz P, Kaiser W, Van Schoot J, Mallmann J, de Pee J, Stoeldraijer J. High-NA EUV lithography optics becomes reality. In Extreme Ultraviolet (EUV) Lithography XI, 11323. SPIE,; 2020. p. 25–35.

[4] Henselmans R, van Drunen C. Flexible and fast non-contact measurement of a large off-axis non-circular AR-coated freeform optic. In Optifab, 11889. SPIE,; 2021. p. 9–14.
[5] Liu S, Wang H, Zhang Q, Hou J, Zhong B, Chen X. Regionalized modeling approach of tool influence function in magnetorheological finishing process for aspherical optics. Optik 2020;206:164368.
[6] Chernyshev A, Chkhalo N, Malyshev I, Mikhailenko M, Pestov A, Pleshkov R, Smertin R, Svechnikov M, Toropov M. Matrix based algorithm for ion-beam figuring of optical elements. Precis Eng 2021;69:29–35.
[7] Zhong B, Deng WH, Chen XH, Zheng N. Precision manufacture of aspheric optics by robot-based bonnet polishing. Second Target Recognition and Artificial Intelligence Summit Forum, 11427. SPIE,; 2020. p. 409–15.
[8] Wang C, Zhang Z, Cheung CF, Luo W, Loh YM, Lu Y, Kong L, Wang S. Maskless fluid jet polishing of optical structured surfaces. Precis Eng 2022;73:270–83.
[9] Li Y, Wang D. Study on distortion control technology of the active stressed lap polishing deeper aspherical mirror. In ICO20: Optical Devices and Instruments, 6024. SPIE,; 2005. p. 450–7.
[10] Ke X, Wang T, Zhang Z, Huang L, Wang C, Negi VS, Pullen WC, Choi H, Kim D, Idir M. Multi-tool optimization for computer controlled optical surfacing. Opt Express 2022;30(10):16957–72.
[11] Cao H, Zhou J, Jiang P, Hon KK, Yi H, Dong C. An integrated processing energy modeling and optimization of automated robotic polishing system. Robot Cim-Int Manuf 2020;65:101973.
[12] Xiao M, Ding Y, Yang G. A model-based trajectory planning method for robotic polishing of complex surfaces. IEEE T Autom Sci Eng 2021;19(4):2890–903.
[13] Pan R, Zhu X, Wang Z, Zhao W, Sun K, Chen D, Fan J. Optimization of static performance for robot polishing system based on work stiffness evaluation. P I Mech Eng B-J Eng 2023;237(4):519–31.
[14] Ding Y, Zhao J, Min X. Impedance control and parameter optimization of surface polishing robot based on reinforcement learning. P I Mech Eng B-J Eng 2023;237 (1-2):216–28.
[15] Huang T, Zhao D, Cao ZC. Trajectory planning of optical polishing based on optimized implementation of dwell time. Precis Eng 2020;62:223–31.
[16] Alam ST, Tomal AA, Nayeem MK. High-speed machining of Ti–6Al–4V: RSM-GA based optimization of surface roughness and MRR. Results Eng 2023;17:100873.
[17] Sharma N, Gupta RD, Khanna R, Sharma RC, Sharma YK. Machining of Ti-6Al-4V biomedical alloy by WEDM: Investigation and optimization of MRR and Rz using grey-harmony search. World J Eng 2023;20(2):221–34.
[18] Li B, Tian X, Zhang M. Modeling and multi-objective optimization method of machine tool energy consumption considering tool wear. Int J Pr Eng Man-gt 2022: 1–5.
[19] Rouniyar AK, Shandilya P. Soft computing techniques for modelling and multi-objective optimization of magnetic field assisted powder mixed EDM process. Neural Comput Appl 2022;34(21):18993–9014.
[20] Osorio-Pinzon JC, Abolghasem S, Maranon A, Casas-Rodriguez JP. Cutting parameter optimization of Al-6063-O using numerical simulations and particle swarm optimization. Int J Adv Manuf Tech 2020;111(9):2507–32.
[21] Storn R. On the usage of differential evolution for function optimization. In Proceedings of North American fuzzy information processing. IEEE,; 1996 Jun 19. p. 519–23.
[22] Chakraborty S, Saha AK, Ezugwu AE, Agushaka JO, Zitar RA, Abualigah L. Differential evolution and its applications in image processing problems: a comprehensive review. Arch Comput Method E 2023;30(2):985–1040.
[23] Brest J, Greiner S, Boskovic B, Mernik M, Zumer V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE T Evol Comput 2006;10(6):646–57.
[24] Tanabe R, Fukunaga A. Success-history based parameter adaptation for differential evolution. IEEE congress on evolutionary computation 2013 Jun 20. IEEE,; 2013. p. 71–8.
[25] Zhang J, Sanderson AC. JADE: adaptive differential evolution with optional external archive. IEEE T Evol Comput 2009;13(5):945–58.
[26] Tanabe R, Fukunaga A. Success-history based parameter adaptation for differential evolution. IEEE congress on evolutionary computation 2013 Jun 20. IEEE,; 2013. p. 71–8.
[27] Awad NH, Ali MZ, Suganthan PN, Reynolds RG. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. IEEE congress on evolutionary computation (CEC) 2016 Jul 24. IEEE,; 2016. p. 2958–65.
[28] Awad NH, Ali MZ, Suganthan PN. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In2017. IEEE congress on evolutionary computation (CEC). IEEE,; 2017 Jun 5. p. 372–9.
[29] Akhmedova S, Stanovov V, Semenkin E. LSHADE Algorithm with a Rank-based Selective Pressure Strategy for the Circular Antenna Array Design Problem. ICINCO 2018;(1):159–65.
[30] Fu T, Liu S, Li P. Digital twin-driven smelting process management method for converter steelmaking. J Intell Manuf 2024:1–7.
[31] Lee J, Su H. A unified industrial large knowledge model framework in smart manufacturing. Int J AI Mater Des 2024;1(2):41–7.
[32] Wang J, Shi Z, Yu P, Wang Z. Predicting the Material Removal Rate in Chemical Mechanical Planarization Based on Improved Neural Network. IEEE Access 2023.
[33] Tian Y, Ma Z, Ma X, Li L, Yan J. A cross-scale material removal prediction model for magnetorheological shear thickening polishing. J Mater Process Tech 2024;332: 118569.
[34] Mirjalili S, Mirjalili S. Genetic algorithm. Evolut Algorithms Neural Netw: Theory Appl 2019:43–55.
[35] Maruyama T, Igarashi H. An effective robust optimization based on genetic algorithm. IEEE T Magn 2008;44(6):990–3.
[36] Ai S, Song J, Cai G. A real-time fault diagnosis method for hypersonic air vehicle with sensor fault based on the auto temporal convolutional network. Aerosp Sci Technol 2021;119:107220.
[37] Rogers A, Prugel-Bennett A. Genetic drift in genetic algorithm selection schemes. IEEE T Evol Comput 1999;3(4):298–303.
[38] Khatib W, Fleming PJ. The stud GA: A mini revolution?. In International Conference on Parallel Problem Solving from Nature. Berlin, Heidelberg: Springer Berlin Heidelberg,; 1998 Sep 27. p. 683–91.
[39] Syswerda G. A study of reproduction in generational and steady-state genetic algorithms. Foundations of genetic algorithms, 1. Elsevier,; 1991 Jan 1. p. 94–101.
[40] Beyer HG, Schwefel HP. Evolution strategies–a comprehensive introduction. Nat Comput 2002;1:3–52.